Programming Languages I Might Like To Learn
Timothy Germakovski

Abstract:
This assignment doesn't actually have any programming, and doesn't deal with the languages we will be studying in this course.
Instead, this is basically a wishlist/Amazon shopping cart of 6 programming languages we might want to learn in the future. This
is supposed to broaden our understanding of what's out there. Nice.

Language 1:
Assembly x86 was developed by Intel in the 70s. It is important because it helps bridge the gap between the human and the computer,
incomprehensible binary and something akin to English. I want to learn assembly because it is lower level and my knowledge is very lacking in that area.
The same way users use programs without understanding what makes them work, I as someone who mainly works in java, take for granted what's under the hood.
It just magically works. For this reason I want to better acquaint myself with assembly code.

Language 2:
C++ is a language invented in 1979 by some Danish guy called Bjarne.
He wanted to make "C with classes", so he took the very popular C and made a higher level version.
He was motivated to do this because Simula was too slow. The name C++ comes from Rick Mascitti. The
plus plus means to increment in C, so C++ means evolved/improved C. I would want to know C++ because I
already know C and this seems to be a nicer version of that.
It is also the third most popular language in the world.

Language 3:
Python was invented in the 80s by Guido to replaces the ABCs, to handle exceptions like Б or ش or 诶 I would presume.
My boi Guido developed Python up until the moment that he stopped developing Python, which was on the twelfth of July,
in the year two thousand and eighteen of our Lord. I was to learn Python because it is even more popular than C++ and also
because my friend keeps telling me its better than Java and C, to which I have no reply because I'm not familiar with Python.

Language 4:
Brainf*ck is an esoteric minimalist language developed in 1992 by some Swiss student named Urban Muller, to mess with people's brains.

It only has 8 characters, which I guess makes it 4 times more complex than binary. Need I say more. This language does not exist for practical
purposes, even though Urban's original goal was to implement the world's smallest compiler. It is just a challenges, and I'm to see how something like
++++++++[>++++[>++>+++>+++>+<<<<-]>+>+>->>+[<]<-]>>.>---.+++++++..+++.>>.<-.<.+++.------.--------.>>+.>++. can amount to anything meaningful.

Language 5:
Javascript was made in '95 by Microsoft because web pages were static and somebody wanted to make them dynamic.
JS is very different from the languages I know because it is used for making front end stuff, not back end. Something like roughly most of
the world's web pages use JS. Maybe 98%. I want to familiarize myself with JS to learn front end stuff. Also, Pantaleev always makes fun of JS and
I can't relate, so this is a matter of urgent concern.

Language 6:
Scala was made in 2001 by Martin Odersky to fix everything people don't like about java. As someone who already doesn't have much of an issue with java, I find this premise interesting. How much better can it be? Well apparently, Scala is both object oriented and functional, so that's pretty cool. The name Scala is a combo of scalable and language, because its meant to grow with the demands of its users. I like Scala's syntax. It seems to have a logic to it, like how a method name can be treated like a variable name upon initialization, with a type and "=" sign.